Remarks

Reconsideration of the above-captioned application is respectfully requested. The objection to the claims has been cured and the Section 112 rejection based on an alleged lack of antecedent basis for "IMS system server" in Claim 1, line 5 appears to be incorrect (see lines 3 and 4 of the claim). The informalities will not be further addressed.

Claims 1-4, 11-14, and 21-24 have been rejected under 35 U.S.C. §103 as being unpatentable over Fortin, USPN 5,528,753 in view of Baer et al., USPN 6,035,303, and Claims 5-10, 15-20, and 25-30 have been rejected under 35 U.S.C. §103 as being unpatentable over Fortin in view of Baer et al. and Chan et al., USPN 6,460,178.

To overcome the rejections, Claim 1 has been amended to recite the limitations of now-canceled Claims 2-4. Claims 1 and 5-30 remain pending.

The fact that Applicant has focussed its comments distinguishing the present claims from the applied references and countering certain rejections must not be construed as acquiescence in other portions of rejections not specifically addressed.

Addressing Fortin first, Fortin is directed to a method for monitoring software programs. Monitoring is a very different thing from the focus of the present independent claims (1, 11, and 21), which seek to address contention for a single exit interface within an IMS system. The claimed IMS system allows only a single instance of a given exit interface to enhance the function of the system and user software. In marked contrast, it appears that the exit routine mentioned in Fortin is for the purposes of collecting performance

statistics, rather than for any purpose for which a user chooses to adapt and customize the system. It is believed that the gravamen of this distinction is contained in the independent claims.

With greater specificity as to Fortin, Fortin modifies the "target" software to receive control before and after target execution. With the presently claimed combination of features, no preexisting software need be modified other than the calling sequence of the "target" exits, to thereby transparently expand this single interface to multiple users without modification to any system or user code. Other than Fortin intercepting from a caller and receiving control back from the target, there is no commonality between the present claims and Fortin.

Indeed, nowhere does Fortin address the topic of enabling a single exit to be used by multiple users. The portions of Fortin (elements 706, 720, and 722 in figure 7 "and associated text" and col. 5, lines 45-60 and col. 6, lines 19-67) that have been relied on for this teaching nowhere mention that multiple users use a single exit, much less hint at anything resembling the resolution of name ambiguity as claimed in the independent claims. Instead, element 706 is a line drawn from the entry section of the text of a stripped object to a common entry code in a library. Elements 720 and 722 are lines drawn between the common exit code in the library and a user-supplied exit routine, strongly implying just the opposite of what is claimed. The relied-upon part of col. 5 of Fortin discloses that an "entry is provided ***for each target routine***", lines 48 and 49, and that a single entry and single exit routine are selected from plural routines, lines 53-55. Both of these teachings are strongly suggestive of an exit/entry being selected for a specific target routine, the opposite of multiple target routines using a single exit. Column 6 of Fortin adds to the teachings away noted above. In any case, nothing in the sections of Fortin that have been relied on for resolving name ambiguity and using a single exit for multiple users without such ambiguity in fact state anything of the sort.

With respect to Baer, it appears that this reference bears little if any relevance to the present claims. The present inventor has reviewed the reference and has declared himself baffled by its import. Indeed, Baer is directed to storing complex objects in digital libraries and nowhere mentions user entry and exit routines, much less that its principles, which are intended to simplify storing and accessing things that are very much different from the routines of Fortin, can be used with such routines. The proferred motivation to combine has been made without regard to what the references fairly teach or suggest.

Turning to Chan et al., it appears that this reference is directed to the construction and optimization of programs rather than control and flow, which is the key point in the present claims. Chan et al appears to be run in preparation of actual program execution, not as an intercept routine that is run as part of the actual program execution.

Very explicit recitations in dependent claims have also been rejected based on citations in the references that seem to state nothing about the subject matter being rejected. By way of non-limiting example only, Claim 8 requires comparing a "candidate user-exit" load module to a predetermined interception routine "eye-catcher", and treating a non-matching "candidate user-exit" load module as a user exit routine, it being alleged that this is taught in Fortin, figures 3 and 7 and "associated text". On the contrary, figure 3 and its "associated text" (col. 4, lines 40-65) simply show and discuss a simple block diagram of how data is collected when a target routine is exited and entered. Figure 7 "and associated text" are of no further avail, because this portion of Fortin simply provides more data gathering details. There is simply no comparison of anything at all in the relied-upon portions of Fortin, much less the particularly recited comparison of, e.g., Claim 8.

The Examiner is cordially invited to telephone the undersigned at (619) 338-8075 for any reason which would advance the instant application to allowance.

Respectfully submitted,

John L. Rogitz
Registration No. 33,549
Attorney of Record
750 B Street, Suite 3120
San Diego, CA 92101
Telephone: (619) 338-8075

JLR:jg

1176-30.AMD